SEMA 4.0 Linux User Manual / Test Procedure

Version 1.2

**Record of Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No.** | **Prepared By** | **Reviewed By** | **Release Date** | **Modifications Done** |
| 1.0 | Hariharan V,  Katha Ashok Kumar, Sankar Raj and  Santhana Kumar | Dhanya Aravind | 30-May-2019 | Initial version |
| 1.1 | Hariharan V,  Katha Ashok Kumar | Dhanya Aravind | 14 June 2019 | Updated Compilation steps.  Added steps to install native GPIO driver.  Updated Backlight access info. |
| 1.2 | Katha Ashok Kumar | Dhanya Aravind | 26 June 2019 | Renamed wdog-test to wdogtest.  Added information for gpio and fan for clear understanding. |

Table of Contents

[1 Getting Started 3](#_Toc11433675)

[2 Compiling the SEMA Linux Package 3](#_Toc11433676)

[3 Installing the SEMA Linux Kernel Drivers 4](#_Toc11433677)

[4 Using the SEMA Linux Module 5](#_Toc11433678)

[4.1 Board Information 5](#_Toc11433679)

[4.2 Voltage Monitor 5](#_Toc11433680)

[4.3 Watchdog 6](#_Toc11433681)

[4.4 GPIO 7](#_Toc11433682)

[4.5 Fan 7](#_Toc11433683)

[4.6 Storage 9](#_Toc11433684)

[4.7 Backlight 10](#_Toc11433685)

[5 Test Scripts 10](#_Toc11433686)

# Getting Started

Install GIT utility to download and compile ADLINK SEMA module for Linux. Install git tool.

$ sudo apt install git

Install “Hexer” for editing the binary files.

$ sudo apt install hexer

Next, install i2c-tools

$ sudo apt install i2c-tools

The current SEMA version was validated on Ubuntu 16.04.6.

# Compiling the SEMA Linux Package

1. Download the source code from ADLINK git repository:

$ git clone <http://GitLab.Adlinktech.com/AATI_CCoE/SEMA_4.0_Linux_SandBox.git>

1. Change directory to workspace.

$ cd SEMA\_4.0\_Linux\_SandBox

1. Compile

$ make

1. After compilation, the following kernel modules and executables will be generated.

|  |  |
| --- | --- |
| **File** | **Description** |
| driver/adl-bmc.ko | SEMA Linux BMC Driver Core Module |
| driver/adl-bmc-boardinfo.ko | SEMA Linux BMC Board Information Driver Module |
| driver/adl-bmc-vm.ko | SEMA Linux BMC Voltage Monitor Driver Module |
| driver/adl-bmc-fan.ko | SEMA Linux BMC Smart Fan Driver Module |
| driver/adl-bmc-wdt.ko | SEMA Linux BMC Watchdog Driver Module |
| driver/adl-bmc-nvmem.ko | SEMA Linux BMC NVMEM Driver Module |
| driver/adl-bmc-bklight.ko | SEMA Linux BMC Backlight Driver Module |
| wdog-test | SEMA Linux Watchdog Test application |
| adltest | SEMA Linux Command Line Utility |

# Installing the SEMA Linux Kernel Drivers

The following commands will install the SEMA related device drivers.

$ cd SEMA\_4.0\_Linux\_SandBox

$ sudo make install

$ sudo modprobe i2c\_i801

$ sudo modprobe adl-bmc

$ sudo modprobe adl-bmc-boardinfo

$ sudo modprobe adl-bmc-vm

$ sudo modprobe adl-bmc-wdt

$ sudo modprobe adl-bmc-fan

$ sudo modprobe adl-bmc-nvmem

$ sudo modprobe adl-bmc-bklight

Install native GPIO driver by using following steps:

1. Load the native GPIO diriver

$ sudo modprobe gpio-pca953x

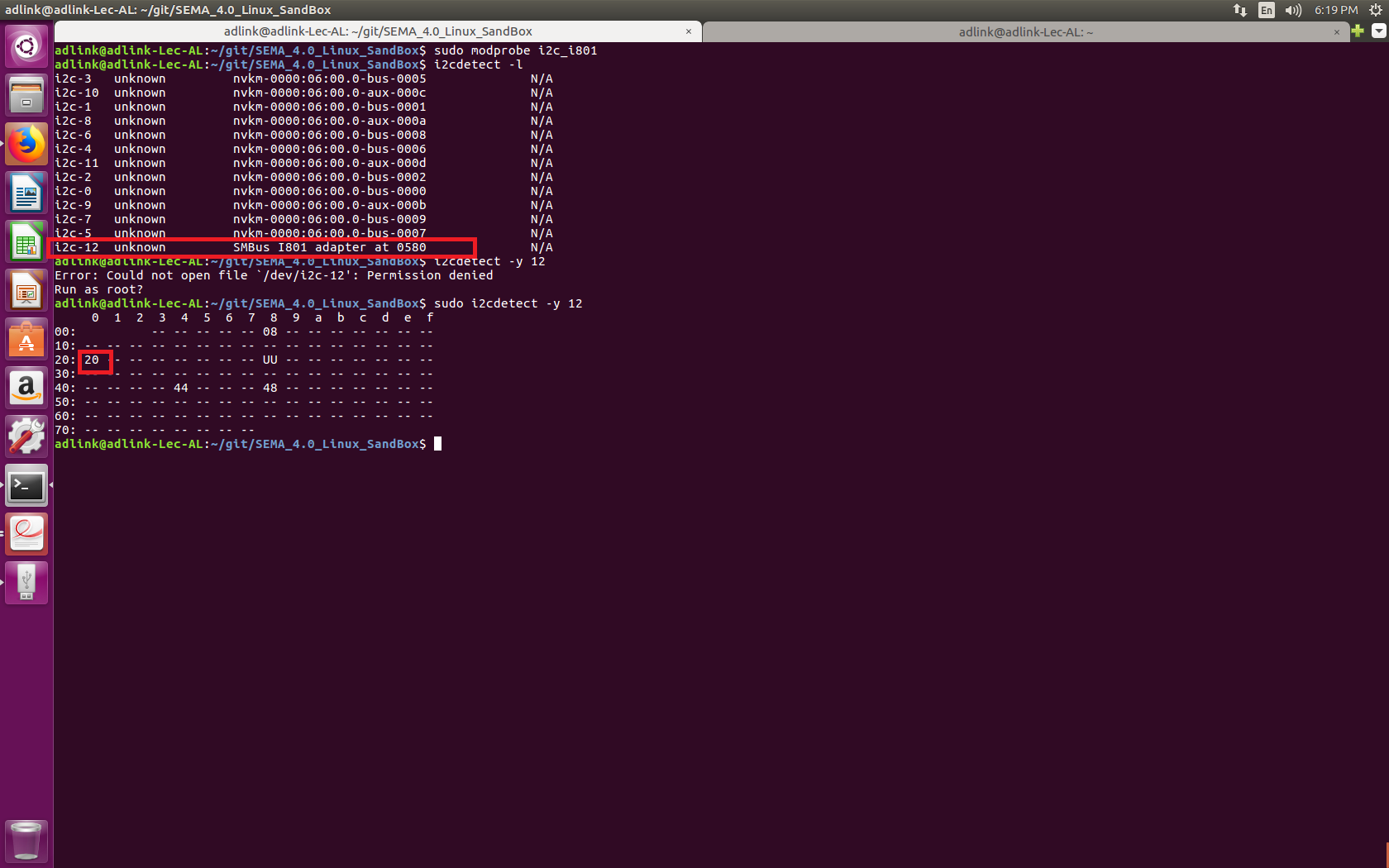
1. After loading kernel driver, configure the GPIO device with the following command.

$ echo pca9535 0x20 > /sys/bus/i2c/devices/i2c-12/new\_device

Here i2c-12 is used, since SMBus is located at bus 12.

0x20 is GPIO device slave address.

Refer the below screenshot to know the i2c bus number and GPIO device address.



# Using the SEMA Linux Module

## Board Information

* To get the board name

**$ cat /sys/bus/platform/devices/adl-bmc-boardinfo/information/board\_name**

* To get current position error log

**$ cat /sys/bus/platform/devices/adl-bmc-boardinfo/information/cur\_pos\_error\_log**

* To set position (read pointer) and get error log immediately

**$ echo 1 > /sys/bus/platform/devices/adl-bmc-boardinfo/information/error\_log**

**$ cat /sys/bus/platform/devices/adl-bmc-boardinfo/information/error\_log**

* To set position (read pointer) and get error number description immediately

**$ echo 1 > /sys/bus/platform/devices/adl-bmc-boardinfo/information/err\_num\_des**

**$ cat /sys/bus/platform/devices/adl-bmc-boardinfo/information/err\_num\_des**

* To get exception description

**$ cat /sys/bus/platform/devices/adl-bmc-boardinfo/information/exc\_des**

* To set the read pointer to a specified exception code number

**$ echo 0 > /sys/bus/platform/devices/adl-bmc-boardinfo/information/exc\_des**

**Note**: LEC-AL Module doesn’t support Error Log feature.

## Voltage Monitor

* To get the hardware monitor input voltage and get the respective description

**$ cat /sys/class/regulator/<regulator index>/microvolts**

**$ cat /sys/class/regulator/<regulator index>/name**

* In the example below, user is reading the voltage and description of id 1.

**$ cat /sys/class/regulator/regulator.1/microvolts**

**$ cat /sys/class/regulator/regulator.1/name**

## Watchdog

* To start the watchdog

**$ echo 1 > /dev/watchdog0**

* To stop the watchdog

**$ echo “V” > /dev/watchdog0**

* To trigger / ping the watchdog

**$ wdogtest –ping**

* To update the watchdog timeout value

To set the watchdog timeout out value, use the following format:

**wdogtest –timeout=<time in seconds>.**

The example below sets the watchdog timeout value as 20 seconds.

**$ wdogtest –timeout=20**

* To get the information about the watchdog state

**$ cat /sys/class/watchdog/watchdog0/state**

* To get the current watchdog timeout value

**$ cat /sys/class/watchdog/watchdog0/timeout**

* To get minimum/maximum watchdog timeout value

**$ /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/wdt\_min\_timeout**

**$ /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/wdt\_max\_timeout**

* To start / update the power up watchdog timer value

To update timeout out value, use the following format:

**echo <timeout> > /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/PwrUpWDog**

The example below sets the power up watchdog timer timeout value as 1000 seconds.

**$ echo 1000 > /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/PwrUpWDog**

* To get the power up watchdog timer value

**$ cat /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/PwrUpWDog**

* To stop the power up watchdog timer

**$ echo 0 > /sys/bus/platform/devices/adl-bmc-wdt/Capabilities/PwrUpWDog**

## GPIO

* To export the GPIO pin

**$ echo N > /sys/class/gpio/export** (N should replace with gpio chip base number)

In below example 251 is the gpio chip base number in LEC-AL.

**Ex: $ echo 251 > /sys/class/gpio/export**

* To read the direction of a pin

**$ cat /sys/class/gpio/gpio251/direction**

* To write the direction of a pin

**$ echo in > /sys/class/gpio/gpio251/direction**

* To read the value of a pin

**$ cat /sys/class/gpio/gpio251/value**

* To write the value of a pin

**$ echo 1 > /sys/class/gpio/gpio251/value**

## Fan

Based on platform (LEC-AL, cExpress-WL, etc) device name (hwmonX) will change. To confirm the device name which is created by our driver, check the **/sys/bus/platform/devices/adl-bmc-fan/hwmon/** path**.**

Ex: $ ls /sys/bus/platform/devices/adl-bmc-fan/hwmon/

Output**: hwmon2**

In the above example our device is hwmon2, so need to replace hwmonX in below commands with hwmon2

* To set temperature set points (1-4)

**echo <temperature> > /sys/class/hwmon/hwmonX/device/pwm<fan ID>\_auto\_point<set point number>\_temp**

In the example below, the user is setting temperature of set point 1 and Fan ID as 1 (SEMA\_EAPI\_ID\_FAN\_CPU) with temperature as 10

**$ echo 10 > /sys/class/hwmon/hwmon2/device/pwm1\_auto\_point1\_temp**

* To get temperature set points (1-4)

**cat /sys/class/hwmon/hwmonX/device/pwm<Fan Id>\_auto\_point<set point number>\_temp**

In the example below, the user is getting temperature of set point 1 and fan ID 1 (SEMA\_EAPI\_ID\_FAN\_CPU)

**$ cat /sys/class/hwmon/hwmon2/device/pwm1\_auto\_point1\_temp**

* To set PWM set points (1-4)

**echo <PWM value> > /sys/class/hwmon/hwmonX/device/pwm<fan ID>\_auto\_point<set point number>\_pwm**

In the example below, user is setting PWM value of set point 1 and fan ID 1 (SEMA\_EAPI\_ID\_FAN\_CPU) with PWM level as 10

**$ echo 10 > /sys/class/hwmon/hwmon2/device/pwm1\_auto\_point1\_pwm**

* To get PWM set points (1-4)

**cat /sys/class/hwmon/hwmonX/device/pwm<fan ID>\_auto\_point<set point number>\_pwm**

In the example below, user is getting PWM value of set point 2 and fan ID as 2 (SEMA\_EAPI\_ID\_FAN\_SYSTEM\_1 ).

**$ cat /sys/class/hwmon/hwmon2/device/pwm2\_auto\_point2\_pwm**

* Setting one of the fan modes below:
  + SEMA\_FAN\_AUTO (0)
  + SEMA\_FAN\_OFF (1)
  + SEMA\_FAN\_ON (2)
  + SEMA\_FAN\_AUTO\_SOFT\_FAN (3)
* To set fan mode

**echo <fan mode> > /sys/class/hwmon/hwmonX/device/pwm1\_enable**

In the example below, user is setting fan mode as 1

**$ echo 1 > /sys/class/hwmon/hwmon2/device/pwm1\_enable**

* To get fan mode

$ **cat /sys/class/hwmon/hwmon2/device/pwm1\_enable**

* Temperature sources:

0 = CPU Temperature

1 = Board Temperature

* To set temperature source

**echo <temperature source> > /sys/class/hwmon/hwmonX/device/pwm1\_auto\_channels\_temp**

In the example below, user is setting temperature source as 1 (Board temperature)

**$ echo 1 > /sys/class/hwmon/hwmon2/device/pwm1\_auto\_channels\_temp**

* To get temperature source

**$ cat /sys/class/hwmon/hwmon2/device/pwm1\_auto\_channels\_temp**

## Storage

* To get the storage capabilities

**$ cat /sys/bus/platform/devices/adl-bmc-nvmem/capabilities/nvmemcap**

* To read the storage area,

**hexdump /sys/bus/nvmem/devices/<nvmem index>/nvmem**

For example:

**$ hexdump /sys/bus/nvmem/devices/nvmem544/nvmem**

* To write the storage area

**hexer /sys/bus/nvmem/devices/<nvmem index>/nvmem**

For example:

**$ hexer /sys/bus/nvmem/devices/nvmem544/nvmem**

Hexer will open a nvmem file in editor mode. To edit the data press **r (replace**) and enter new data then press escape and **:wq** to save. Then read the same file using hexdump to check saved changes.

## Backlight

* To get the current Backlight Enable State

**$cat /sys/class/backlight/adl-bmc-bklight/bl\_power**

* To Set the current Backlight Enable State to either ON / OFF

**$ echo <enable> > /sys/class/backlight/adl-bmc-bklight/bl\_power**

For Example,

To enable the backlight,

**$ echo 1 > /sys/class/backlight/adl-bmc-bklight/bl\_power**

To Disable the backlight,

**$ echo 0 > /sys/class/backlight/adl-bmc-bklight/bl\_power**

* To get the current Backlight Brightness level

**$cat /sys/class/backlight/adl-bmc-bklight/actual\_brightness**

* To Set the current Backlight brightness level

**$ echo <level> > /sys/class/backlight/adl-bmc-bklight/brightness**

For Example, to set the brightness level of 78,

**$ echo 78 > /sys/class/backlight/adl-bmc-bklight/brightness**

# Test Scripts

Following are the test scripts for testing individual drivers.

Before running the script change its permissions.

**$ sudo chmod a+x gpiotest.sh**

|  |  |
| --- | --- |
| **Driver** | **Test Script** |
| adl\_bmc\_boardinfo | boardtest.sh |
| adl\_bmc\_vm | vmtest.sh |
| adl-bmc-fan | fantest.sh |
| adl-bmc-nvmem | nvmemtest.sh |
| gpio-pca953x | gpiotest.sh |

**Note: All the commands require administrator privileges and run as root user.**